

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-149001

(43)Date of publication of application : 13.06.1995

(51)Int.CI.

B41J 5/30
G06F 3/12

(21)Application number : 05-320972

(71)Applicant : MELCO:KK

(22)Date of filing : 26.11.1993

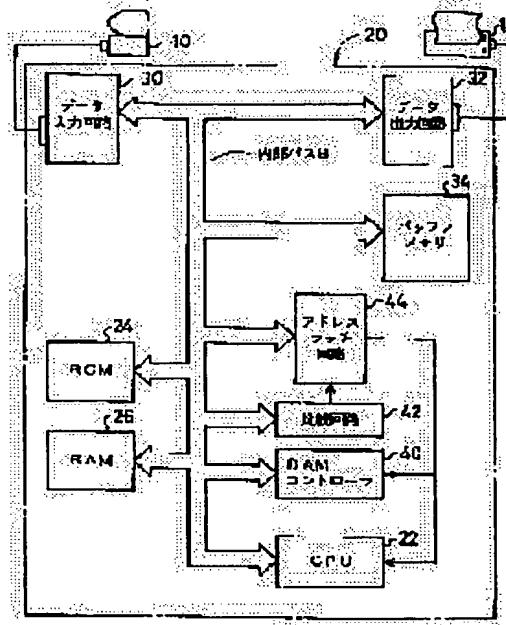
(72)Inventor : TSUZUKI CHIKAU

(54) PRINTER BUFFER

(57)Abstract:

PURPOSE: To store the print data inputted from a computer in DMA and to process the control command contained in the data.

CONSTITUTION: A CPU 22 clears an address latch circuit 44 or the use of the DMA region of a buffer memory 34 to set the command head data of a control command to a comparison circuit 42. Next, the CPU 22 designates the number of DMA transmission bites to a DMA controller 40. When the DMA controller 40 starts DMA processing, the print data inputted from a data input circuit 30 is directly transmitted to the buffer memory 34 and, when the data coinciding with the command head data appears on an internal bus B during the processing of the print data, a latch trigger signal is outputted to the address latch circuit 44 from the comparison circuit 42 and the address data of the buffer memory 34 having the command head data stored therein is successively stored in the address latch circuit 44.



LEGAL STATUS

[Date of request for examination] 10.05.1996

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2835487

[Date of registration] 09.10.1998

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-149001

(43)公開日 平成7年(1995)6月13日

(51)Int.Cl.⁶

B 41 J 5/30

G 06 F 3/12

識別記号

府内整理番号

F I

技術表示箇所

Z

B

審査請求 未請求 請求項の数1 FD (全7頁)

(21)出願番号

特願平5-320972

(22)出願日

平成5年(1993)11月26日

(71)出願人 390040187

株式会社メルコ

愛知県名古屋市中区大須4丁目11番50号

(72)発明者 都築 誠

名古屋市中区大須四丁目11番50号 株式会
社メルコ内

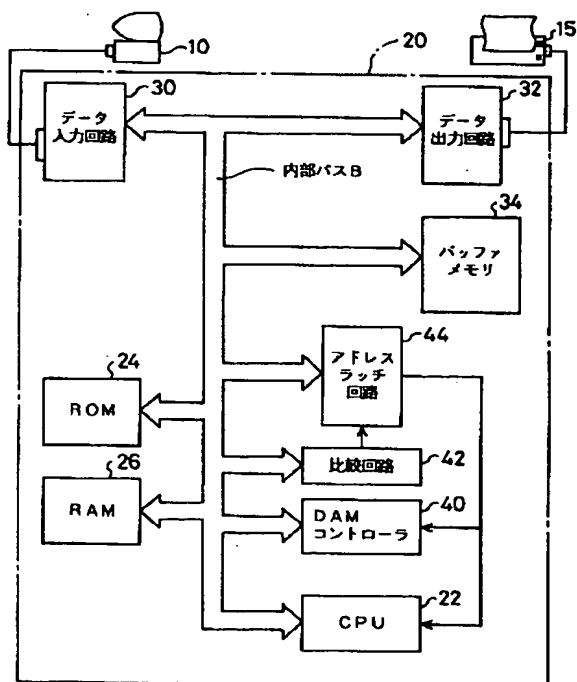
(74)代理人 弁理士 下出 隆史 (外1名)

(54)【発明の名称】 プリンタバッファ

(57)【要約】

【目的】 コンピュータから入力されるプリントデータをDMAにより蓄積すると共にそのデータに含まれる制御コマンドを処理する。

【構成】 CPU 22はアドレスラッチ回路44やバッファメモリ34のDMA領域使用をクリアし、比較回路42に制御コマンドのコマンドヘッドデータをセットする。次にCPU 22は、DMAコントローラ40に対してDMA転送バイト数などを指定する。DMAコントローラ40がDMA処理を開始すると、データ入力回路30から入力されたプリントデータは直接バッファメモリ34へ転送され、この処理の最中にコマンドヘッドと一致するデータが内部バスBに現われると、比較回路42からアドレスラッチ回路44へラッチトリガ信号が出力され、そのコマンドヘッドが記憶されたバッファメモリ34のアドレス情報がアドレスラッチ回路44に順次記憶される。



【特許請求の範囲】

【請求項1】 プロセッサからのダイレクトメモリアクセス指令に従って、コンピュータから出力されたプリントデータを前記プロセッサの動作と非同期にバッファメモリに蓄積するプリンタバッファにおいて、前記プロセッサの実行すべき制御コマンドを記憶するコマンド記憶部と、

該コマンド記憶部に記憶された制御コマンドと前記バッファメモリにダイレクトメモリアクセス方式により転送される前記プリントデータとが一致したとき、該プリントデータが記憶される前記バッファメモリのアドレスを記憶するアドレス記憶手段と、

前記バッファメモリに記憶されたプリントデータを参照して、前記アドレス記憶手段に記憶されたアドレスのプリントデータを読み出し、前記制御コマンドに応じた処理を実行する制御実行手段とを備えることを特徴とするプリンタバッファ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、プロセッサからのダイレクトメモリアクセス（以下、DMAという）指令に従って、コンピュータから出力されたプリントデータを前記プロセッサの動作と非同期にバッファメモリに蓄積するプリンタバッファに関する。

【0002】

【従来技術】 従来、プリンタバッファは、コンピュータから送信されるプリントデータをより高速に蓄積するために、内部プロセッサの動作と非同期にプリントデータをプリンタバッファに記憶するDMA方式を採用している。また、高機能のプリンタバッファは、予め定められた制御コマンドを受け取ると、その制御コマンドを解読および実行することでプリントに関する各種の機能、例えばプリントデータを出力するプリンタの選定や優先順位の指定、プリントの部数、プリントデータの消去などを実行する。このため利用者は、印字データに先だって、あるいは印字データと共に、コンピュータから制御コマンドを送信するだけで、プリンタバッファを遠隔操作することができる。

【0003】 この様に高機能化したプリンタバッファにおいては、コンピュータから送信されるプリントデータにプリンタへ出力すべき情報とプリンタバッファ内部で処理すべき制御コマンドの情報とが混在する。このためプリンタバッファは、プリントデータをDMA方式によりバッファメモリに転送記憶している最中に、制御コマンドを切り出すハードウェアを搭載している。DMA方式によりプリントデータをバッファメモリに記憶しているときには、プリンタバッファのプロセッサは内部バスから切り離された状態にあり、プリントデータの内容を読み込むことはできないからである。従って、プロセッサ以外に何等かの情報処理回路を設ける必要があった。

【0004】 この要求に応えるために、本出願人は特開平4-68433号に開示されるプリンタバッファを提案している。この技術によれば、プリンタバッファに、プリントデータ中に混在する制御コマンドのデータを検出する手段を設け、この手段が制御コマンドを検出すると、プロセッサに対してDMA処理の中断および検出された制御コマンドの実行を要求する割込みを発生する。従って本技術により、DMA処理によるプリントデータの高速記憶と制御コマンドの解読および実行によるプリント作業の高機能化とを両立させた優れたプリンタバッファが提供される。

【0005】

【発明が解決しようとする課題】 しかしながら、近年、プリンタの処理能力が飛躍的に向上し、プリンタバッファに更に高速なプリントデータの処理を実行させたいとする要求が生じると、割り込み方式による処理では、十分に対応できない技術的な問題が見いだされた。前回提案したプリンタバッファは、DMA処理中に制御コマンドが検出されるとプロセッサに対して割込みを発生し、DMA処理の中断と制御コマンドの実行を要求するものである。従って、プリントデータ中に多数の制御コマンドが含まれている場合には、高速なデータ転送を実現するためのDMA処理に中断が多発し、プリントデータの転送速度が低下してしまう。

【0006】 また、上記プロセッサに対する割込み処理には、割り込み信号の発生から実際にこれが受け付けられるまでに相当の時間が必要であり、その処理時間内にコンピュータから出力されたプリントデータを取りこぼしてしまう可能性があった。例えば、コンピュータとプリンタバッファとの間のプリントデータ転送がセントロニクス規格などのようにハンドシェイクを行なう通信規格に則って実行される場合ならば、上記のごとき割込み処理中にはコンピュータからのプリントデータ出力に対して応答しないため、データ転送速度の低下だけが問題となる。しかし、シリアル通信などのようにプリントデータの通信がハンドシェイク無しに実行される場合には、割込み処理中にプリントデータがオーバーランする可能性があり、プリントデータの取りこぼし発生の可能性があった。

【0007】 特に、最近のOA環境にあっては、コンピュータとプリンタバッファとの通信距離が長くなるLAN構築の要求が大きく、シリアル通信の需要は旺盛で、かつ、その転送速度も漸次向上しており、上記問題の解決が望まれていた。

【0008】 本発明のプリンタバッファは、こうした問題点を解決し、高速化および高度化の一途を辿る利用者の要求を満足する高機能かつ高速処理のプリンタバッファを提供することを目的としてなされ、次の構成を採った。

【0009】

【課題を解決するための手段】本発明のプリンタバッファは、プロセッサからのダイレクトメモリアクセス指令に従って、コンピュータから出力されたプリントデータを前記プロセッサの動作と非同期にバッファメモリに蓄積するプリンタバッファにおいて、前記プロセッサの実行すべき制御コマンドを記憶するコマンド記憶部と、該コマンド記憶部に記憶された制御コマンドと前記バッファメモリにダイレクトメモリアクセス方式により転送される前記プリントデータとが一致したとき、該プリントデータが記憶される前記バッファメモリのアドレスを記憶するアドレス記憶手段と、前記バッファメモリに記憶されたプリントデータを参照して、前記アドレス記憶手段に記憶されたアドレスのプリントデータを読み出し、前記制御コマンドに応じた処理を実行する制御実行手段と、を備えることを特徴とする。

【0010】

【作用】以上のように構成された本発明のプリンタバッファは、コマンド記憶部に記憶された制御コマンドとバッファメモリにダイレクトメモリアクセス方式により転送されるプリントデータとが一致したとき、そのプリントデータが記憶されるバッファメモリのアドレスをアドレス記憶手段が記憶する。そして、制御実行手段は、バッファメモリに記憶されたプリントデータを参照し、アドレス記憶手段に記憶されたアドレスのプリントデータを読み出す。このプリントデータは制御コマンドなので、この制御コマンドに応じた処理を実行する。

【0011】なお、ここでアドレス情報とは、バッファメモリの絶対アドレス情報であっても、DMA処理のページ情報などにより修飾される相対アドレス情報の何れでもよい。

【0012】また、制御コマンドの検出には、コマンド記憶部に記憶された制御コマンドとプリントデータとの完全一致のみならず、制御コマンド体系に共通の情報、例えばコマンドヘッド等の情報とプリントデータとの一致を検出するなど、一般的のデータ検索手法が適宜適用される。

【0013】

【実施例】以上説明した本発明の構成、作用を一層明らかにするために、以下本発明のプリンタバッファの好適な実施例について説明する。図1は、実施例であるプリンタバッファ20の電気回路ブロック図である。図示するようにプリンタバッファ20は、マイクロプロセッサ（以下、CPUという）22を中心とする論理回路により構成され、後述するプログラムなどのデータを不揮発的に記憶するROM24、このROM24と一緒にになってCPU22の主記憶となるRAM26および各種入出力回路などから構成され、これらの回路は互いに内部バスBにより接続されている。なお、この内部バスBは、メモリ関連の信号としてアドレス、読み取りデータ、書き込みデータ、メモリ素子制御の各種信号、また入出力回路

関連の信号として機器選択、入力データ、出力データ、機器制御の各種信号が伝送される信号線である。なお、こうしたバスは、信号線の増加を抑えるためにこれらの信号を時分割で転送することも差し支えない。

【0014】データ入力回路30は、コンピュータ10との間でプリントデータの授受を行なう入出力インターフェイスであり、各種のプリンタ通信規格に則りコンピュータから送信されてくるプリントデータをCPU22の扱えるデータ形式に変更し、適宜タイミングで内部バスBに出力する。データ出力回路32は、上記データ入力回路30の動作とは逆に、CPU22にて処理されたデータを各種のプリンタ通信規格に適合するデータ形式に変更し、プリンタ15へと出力する。また、バッファメモリ34は、大容量の記憶素子から構成されるもので、データ入力回路30から入力されたプリントデータを一時に蓄積し、これをデータ出力回路32に適宜出力する。

【0015】実施例のプリンタバッファ20は、データ入力回路30から入力されたプリントデータをバッファメモリ34に記憶するに際し、公知のDMA処理を実行してデータの転送速度を大幅に改善している。このDMA処理を制御するため上記内部バスBには、DMAコントローラ40が接続されている。

【0016】本実施例のDMAコントローラ40は、いわゆるスチールモードにより内部バスBを制御する。すなわち、DMA処理が必要になると、CPU22はDMA転送の要求およびデータ転送に必要な情報としてDMA転送バイト数（実施例では4Kバイト）、記憶ページの大きさやバッファメモリ34の転送開始番地等を設定する。その後にDMAコントローラ40は、データ入力回路30からプリントデータの入力に応じてCPU22が内部バスBを使用していない期間を利用して、前もつて設定されている転送バイト数および転送開始番地に従ってデータ入力回路30から入力されたプリントデータをバッファメモリ34へ転送するのである。

【0017】更に、本実施例のプリンタバッファ20は、比較回路42およびアドレスラッチ回路44を備えている。比較回路42とは、CPU22からセットされた基準データとDMA処理によりバッファメモリ34に転送されるデータとを比較し、両データが一致したときにアドレスラッチ回路44へラッチトリガ信号を出力する回路である。このラッチトリガ信号を入力するアドレスラッチ回路44は、ラッチトリガ信号が入力されたときの内部バスBのアドレス情報をラッチする回路であり、そのアドレス情報を先入れ先出し（以下、FIFOという）方式により16個記憶する。また、アドレスラッチ回路44のオーバーフロー端子OFは、前記CPU22の割込み端子およびDMAコントローラ40のDMA停止端子に接続されており、記憶容量を越える16個

以上のアドレス情報が入力されたときにはこのオーバーフロー端子OFから信号を出力し、DMA処理を停止させると共にCPU22にオーバーフロー割込みプログラムを実行させる。なお、通常の使用では、印字データ中の制御コマンドの発生頻度が多くても、CPU22が未処理のコマンドを16個以上ためてしまうことは、極めて可能性が低いので、 FIFOがオーバーフローすることはほとんど生じない。なお、多数の制御コマンドが予測されるような場合には、FIFOを更に設ければ良い。

【0018】以上のように構成された本実施例のプリンタバッファ20は、ROM24に下記のごとき各種プログラムが予め記憶されており、次の様に動作する。図2は、DMA要求および実行プログラムのフローチャートであり、CPU22がDMAコントローラ40に内部バスBを使用したDMA処理を要求する際、およびDMA転送の実行中に行なわれるものである。このプログラムの処理を開始するとCPU22は、アドレスラッチ回路44やバッファメモリ34のDMA領域使用をクリアして以後のDMA処理に備える(ステップ100)。

【0019】続いてCPU22は、比較回路42に対して基準データとして2バイトのデータ「@B」をセットする(ステップ110)。ここで、2バイトのデータ「@B」とは、本実施例のプリンタバッファ20に対する各種コマンドの先頭に付加される共通のデータ、いわゆるコマンドヘッドのデータである。すなわち、本実施例のプリンタバッファ20に対してプリント部数を指定したり、プリント作業を希望するプリントを指定などの制御コマンドは、コマンドヘッドの2バイトデータ「@B」に続けて記述されるのである。

【0020】こうしてDMA処理のための総ての準備が完了するとCPU22は、DMAコントローラ40に対してDMAのための初期設定情報として転送バイト数、バッファメモリ34の転送開始番地などを指示し(ステップ120)、スチールモードでのDMA転送を開始させる。DMA転送が始まると、CPU22は、ステップ130以下の処理を実行するが、この処理については、DMA転送による動作の説明の後で詳しく説明する。

【0021】DMA要求に基づいてDMAコントローラ40がDMA処理を開始すると、通常のスチールモードによるDMA転送が実行され、データ入力回路30から入力されたプリントデータはCPU22と非同期に、直接バッファメモリ34へ転送される。そして、このDMA処理の最中にDMA転送されるプリントデータ中に比較回路42にセットされたコマンドヘッド「@B」と一致するデータが内部バスBに現われたとき、比較回路42からアドレスラッチ回路44へラッチトリガ信号が出力され、そのコマンドヘッド「@B」が記憶されたバッファメモリ34のアドレス情報がアドレスラッチ回路44にFIFO方式により順次記憶される。すなわち、C

PU22を全く介在させることなく、完全にハード的に、コマンドヘッド「@B」が記憶されたバッファメモリ34のアドレス情報がアドレスラッチ回路44に蓄積される。なお、DMA転送中のアドレス情報が、転送ページ内のローカルアドレスとなっている場合には、上位のアドレスを設定する際、これを所定のアドレスに記憶しておき、後で参照するものとすれば良い。もとより、上位アドレスをFIFOに併せて記憶する構成としてもよい。

10 【0022】DMA転送により以上の処理が行なわれる間、CPU22は、FIFOにデータがセットされているか否かの判断を行なう(ステップ130)。FIFOが空でなければ、FIFOのデータ、即ちソフトウェアコマンドが保存されているアドレスデータを読み出し(ステップ140)、そのアドレスのデータを読み出す(ステップ150)。次に、このデータを解析し、バッファの入力側で処理すべきコマンドであるか否かの判断を行ない(ステップ160)、入力側で処理すべきデータである場合には、この制御コマンドを実行する(ステップ170)。ソフトウェアコマンドには、入力側で実行すべき処理と出力側で実行すべき処理が存在する。入力側で実行すべき処理とは、例えば現在印字中のデータを含めてバッファの内容を直ちにクリアする処理などである。バッファのクリアは、データの区切りポイント以降をクリアする等の処理も存在する。この場合には、先のソフトウェアコマンドを、バッファの所定の領域に保存しておけば良い。出力側で実行すべき処理は、例えば印字先であるプリンタの切替などがある。

20 【0023】制御コマンドを実行した後、あるいはFIFOにデータがセットされていないとき(ステップ130)、セットされたコマンドが入力側で実行すべきコマンドでない場合(ステップ160)には、特に制御コマンドの処理を行なわず、DMA転送が完了したか否かの判断を行ない(ステップ180)、DMA転送の継続中は、ステップ130以下の処理を繰り返す。

30 【0024】次に、バッファメモリ34に蓄積した印字データをプリントアウトする処理について、図3のフローチャートを用いて説明する。このプログラムの処理に入るとCPU22は、DMA処理によりバッファメモリ34に格納された一連のプリントデータの記憶領域を確認する(ステップ200)。この処理により、CPU22のアドレスカウンタにはデータを読み出す先頭アドレスがセットされる。続いて、そのアドレスカウンタに指示された領域からデータを読み出す処理を行なう(ステップ205)。

40 【0025】続いてCPU22は、読み出したデータがソフトウェアコマンドであるか否かの判断を行なう(ステップ210)。ソフトウェアコマンドではない、即ち印字データであると判断された場合には、そのデータを、データ出力回路32へ出力する通常のプリント処理

を実行する（ステップ220）。

【0026】一方、ステップ210にて、読み出したデータがソフトウェアコマンドであると判断された場合には、そのコマンドが出力側で処理すべきコマンドであるか否かの判断を行なう（ステップ225）。出力側で処理すべきコマンド、例えば印字先であるプリンタの切替コマンドである場合には、コマンドに応じた制御を実行する（ステップ230）。

【0027】こうしてアドレスカウンタにて指示されたバッファメモリ34の記憶データの処理が完了すると、アドレスカウンタの内容をインクリメントし（ステップ240）、今回のデータ読み出し範囲総てについて同様の処理が完了するまで前記ステップ205へと戻る。

【0028】この様に、バッファメモリ34にDMA転送されたデータは順次解析され、出力時に実行を要するソフトウェアコマンドの場合には、対応する処理を実行し、それ以外のデータは印字データと判断してデータ出力回路32からプリンタへと出力する。従って、DMA処理によりプリンタへ出力する情報と制御コマンドの情報とが混在されてバッファメモリ34へ格納されるが、その混在データの読み出しに当たっては、簡単かつ確実に制御コマンドを分離、抽出することができ、コンピュータ10からデータ入力回路30を介して入力された制御コマンドに応じた制御をCPU22にて実行することが可能となる。

【0029】図4は、上記のごとく本実施例において重要な役割を果たすアドレスラッチ回路44にオーバーフローが発生し、CPU22に対して割込みが発生したときに実行される割込みプログラムのフローチャートである。なお、通常コンピュータから入力されるデータに含まれる制御コマンドの発生頻度では、CPU22が16個以上のコマンドを未処理のままでしまうことは極めて可能性が低いので、本実施例のように16個のアドレスデータをラッチ可能なアドレスラッチ回路44にオーバーフローが発生する可能性は極めて低い。

【0030】前述のごとくアドレスラッチ回路44は、このオーバーフロー割込みと同時にDMAコントローラ40に対してDMA処理の停止を要求しており、内部バスBの制御はCPU22に委ねられている。そこでCPU22は、データ入力回路30に対してプリントデータの入力状況を確認し、未だにコンピュータからプリントデータが伝送されているか否かを判定し、プリントデータの授受がハンドシェイクにより実行されているか否かを確認する（ステップ300）。すなわち、コンピュータ10とプリンタバッファ20との間のプリントデータの授受がハンドシェイク無しに実行されている場合には、DMA処理の停止とは無関係にコンピュータから一方的にプリントデータが転送されてくる状況が継続している。一方、コンピュータとプリンタバッファ20との間のプリントデータの授受がハンドシェイクにより実行

されている場合には、データ入力回路30へのプリントデータ入力はコンピュータ処理により中断される。

【0031】そこで、次のステップ310によりハンドシェイクの有無を判定し、ハンドシェイクが取られてプリントデータの授受が実行されていたと判断したときには、DMA処理を行なうことなく以後のプリントデータの入力をCPU22が介在したプログラム制御により実行し、入力されるプリントデータにコマンドヘッド「@B」が含まれていた場合にはその情報を検出してRAM26の所定領域に振り分けつつプリントデータをバッファメモリ34へと記憶する（ステップ320）。

【0032】一方、ステップ310にてプリントデータがハンドシェイク無く入力されていた場合でも、入力側で取りこぼされた制御コマンドを印字データ出力側のプログラム（図3）で解析するので、コマンドに対する応答の遅れはあるものの、印字データやコマンドの実行に不具合を生じることはない。

【0033】上述した割込みプログラムによりプリンタバッファ20は、例えアドレスラッチ回路44にデータのオーバーフローが発生したとしても、プリントデータの授受に問題発生の可能性のないハンドシェイク状態であれば以後のプリントデータを受け入れることが可能となる。また、プリンタバッファ20は、プリントデータのオーバーランの可能性がある場合にはその旨をコンピュータへ出力し、自動的にプリントデータの入力を停止することができる。

【0034】以上説明したように本実施例のプリンタバッファによれば、僅か2バイトのデータ比較を高速なハード処理にて行なって制御コマンドを検出するため、DMA処理と同時進行でプリントデータに含まれる制御データを確実かつ正確に分離、処理することができる。従って、一旦開始されたDMA処理を中断することなくプリントデータをバッファメモリ34に入力して最高速のDMA処理が実現されることは勿論のこと、分離された制御コマンドを確実に実行することでプリンタバッファ20を遠隔操作する高機能化の要求をも満足することができる。すなわち、プリンタバッファ20は、CPU22の割込み処理などのように相当の処理時間を必要とするプログラム制御を実行せずにプリントデータを入力することができ、ハンドシェイクの無い高速シリアル通信によりプリントデータが入力される場合にも十分な応答速度をもって対応することができる。

【0035】また、仮にアドレスラッチ回路44に記憶されるアドレス情報がオーバーフローした場合であっても、プリントデータの入力を継続し無駄のないデータ通信を実現することができる。

【0036】なお、本実施例では処理を単純化、高速化するために制御コマンドのコマンド体系を統一して比較回路42において2バイトのコマンドヘッド「@B」を検出しているが、この様な比較回路を複数設けることで

この様なコマンド体系に属し無い別種の制御コマンドを抽出することもできる。

【0037】また、アドレスラッチ回路44にてアドレス情報のオーバーフローが発生した場合の処理を単純化するために、この様な場合には常にDMA処理の中止し、コンピュータにプリントデータのオーバーラン発生を伝えるメッセージを送信してもよい。本実施例では、制御コマンドの解析の一部は、DMA転送時に行なったが、印字データのDMA転送が終了した後にまとめて行なうものとし、解析の簡略化を図ることも差し支えない。

【0038】以上本発明の実施例について説明したが、本発明はこうした実施例に何等限定されるものではなく、スチールモード以外のDMA転送を用いた構成、制御コマンドのヘッダが3バイト以上の構成を備えるものなど、本発明の要旨を逸脱しない種々なる態様により具現化されることは勿論である。

【0039】

【発明の効果】以上説明したように本発明のプリンタバッファは、コマンド記憶部に記憶された制御コマンドとDMA方式によりバッファメモリに転送されるプリントデータとが一致したとき、そのプリントデータが記憶されるバッファメモリのアドレスをアドレス記憶手段が記憶する。そして、バッファメモリに記憶されたプリントデータを参照して、制御実行手段はそのアドレス記憶手段に記憶されたアドレスのプリントデータを読出す。このプリントデータは制御コマンドであると判断できるから、その制御コマンドに応じた処理を実行する。従つて、本発明のプリンタバッファは、DMA処理により最

高速にプリントデータをバッファメモリに格納することは勿論のこと、制御コマンドに応じた複雑な処理を確実に実行し、高機能かつ高速処理のプリンタバッファとして機能する。

【図面の簡単な説明】

【図1】本発明の一実施例であるプリンタバッファの電気回路ブロック図である。

【図2】そのプリンタバッファのCPUにて実行されるDMA要求プログラムのフローチャートである。

【図3】そのプリンタバッファのCPUにて実行されるプリントプログラムのフローチャートである。

【図4】そのプリンタバッファのCPUにて実行される割込みプログラムのフローチャートである。

【符号の説明】

10…コンピュータ

15…プリンタ

20…プリンタバッファ

22…CPU

24…ROM

26…RAM

30…データ入力回路

32…データ出力回路

34…バッファメモリ

40…DMAコントローラ

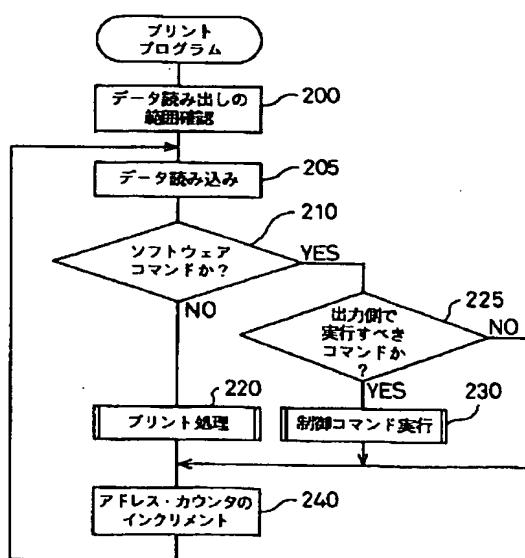
42…比較回路

44…アドレスラッチ回路

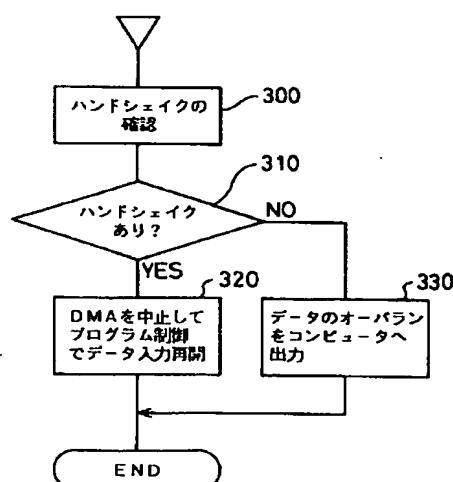
B…内部バス

O F…オーバーフロー端子

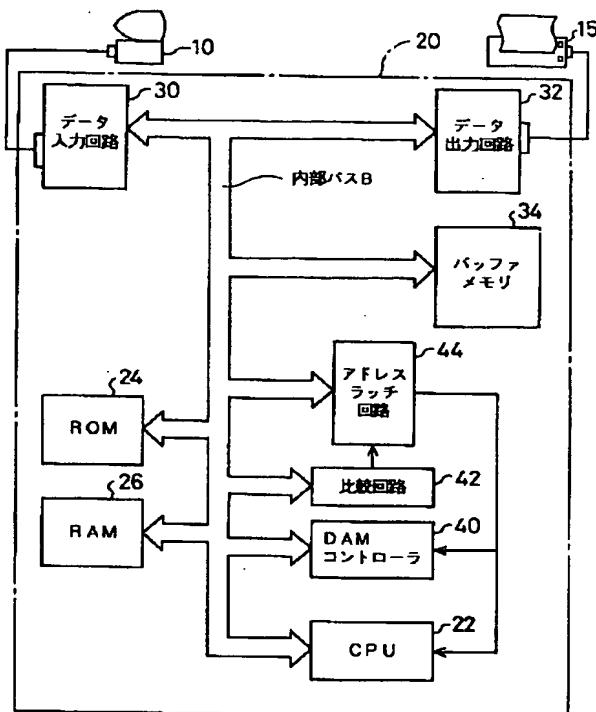
【図3】



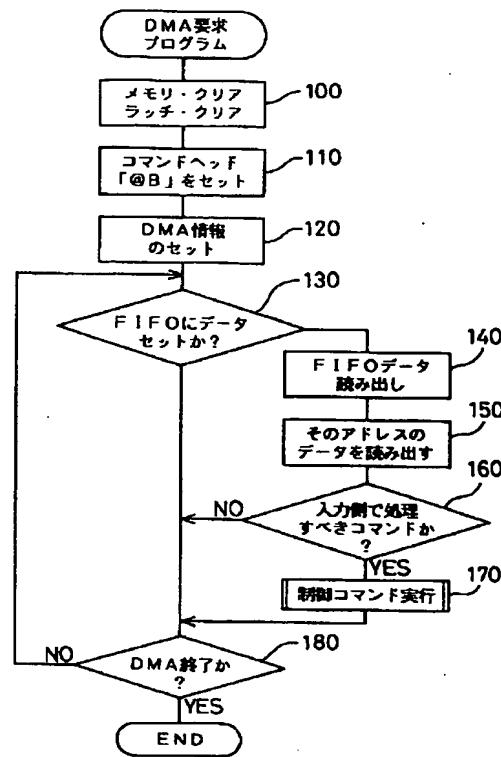
【図4】



【図1】



【図2】



*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The printer buffer which accumulates the print data which are characterized by providing the following, and which were outputted from the computer according to the direct-memory-access instructions from a processor to asynchronous [of the aforementioned processor / operation and asynchronous] at buffer memory. The command storage section which memorizes the control command which should perform the aforementioned processor. An address storage means to memorize the address of the aforementioned buffer memory these print data are remembered to be when the control command memorized by this command storage section and the aforementioned print data transmitted to the aforementioned buffer memory by the direct-memory-access method are in agreement. A control execution means to read the print data of the address memorized by the aforementioned address storage means with reference to the print data memorized by the aforementioned buffer memory, and to perform processing according to the aforementioned control command.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] this invention relates to the printer buffer which accumulates the print data outputted from the computer to asynchronous [of the aforementioned processor / operation and asynchronous] at buffer memory according to the direct-memory-access (henceforth DMA) instructions from a processor.

[0002]

[Description of the Prior Art] Conventionally, the printer buffer has adopted as asynchronous [of an internal processor / operation and asynchronous] the DMA system which memorizes print data to a print buffer, in order to accumulate more the print data transmitted from a computer at high speed. Moreover, a highly efficient printer buffer will perform elimination of various kinds of functions about a print, for example, selection of the printer which outputs print data and specification of priority, number of copies of a print, and print data etc. by decoding and executing the control command, if the control command defined beforehand is received. For this reason, in advance of printing data, with printing data, a user can only transmit control command from a computer, and can operate a printer buffer by remote control.

[0003] Thus, in the high performance printer buffer, the information which should be outputted to a printer, and the information on the control command which should be processed inside a printer buffer are intermingled to the print data transmitted from a computer. For this reason, the printer buffer carries the hardware which starts control command in the midst which is carrying out the transfer storage of the print data by DMA system at buffer memory. It is because the processor of a printer buffer is in the state where it was separated from the internal bus and the contents of print data cannot be read, when having memorized print data to buffer memory by DMA system. Therefore, a certain information processing circuit needed to be prepared in addition to the processor.

[0004] In order to meet this demand, these people have proposed the printer buffer indicated by the publication number No. 68433 [four to]. If according to this technology a means to detect the data of the control command intermingled in print data is prepared in a printer buffer and this means detects control command, the interruption which requires execution of the control command by which DMA processing was interrupted and detected to the processor will be generated. Therefore, the outstanding printer buffer which reconciled high-speed storage of the print data based on DMA processing, decode of control command, and advanced features of the print work by execution with this technology is offered.

[0005]

[Problem(s) to be Solved by the Invention] However, the throughput of a printer improved by leaps and bounds, and when the demand given to a printer buffer wanting to perform processing of still more nearly high-speed print data arose, the technical problem which cannot fully be coped with was found out by processing by the interruption method in recent years. If control command is detected during DMA processing, the printer buffer proposed last time will generate interruption to a processor, and will require discontinuation of DMA processing and execution of control command. Therefore, when much control command is contained in print data,

discontinuation will occur frequently to the DMA processing for realizing high-speed data transfer, and print data transfer speed will fall.

[0006] Moreover, considerable time is required for interrupt processing to the above-mentioned processor by this actually being received from generating of an interrupt signal, and the print data outputted from the computer in the processing time may have been taken and spilt. For example, if it becomes when print data transfer between a computer and a printer buffer is performed in conformity with the telecommunications standard which performs a handshake like a Centronics interface, in order not to answer to the print data output from a computer during interrupt processing like the above, only the fall of a data transfer rate poses a problem. However, when communication of print data was performed without a handshake like serial communication, print data might overrun, and print data took, it spilt, and the possibility of generating was during interrupt processing.

[0007] If it was in the latest OA environment especially, the demand of the LAN construction to which the communication range of a computer and a printer buffer becomes long was large, and it was full of the need of serial communication, and the transfer rate was also improving gradually, and solution of the above-mentioned problem was desired.

[0008] The printer buffer of this invention solved such a trouble, was made for the purpose of offering the high efficiency which satisfies a demand of the user improvement in the speed and an advancement of whom are enhanced, and the printer buffer of high-speed processing, and took the next composition.

[0009]

[Means for Solving the Problem] The printer buffer of this invention follows direct-memory-access instructions from a processor. In the printer buffer which accumulates the print data outputted from the computer to asynchronous [of the aforementioned processor / operation and asynchronous] at buffer memory The command storage section which memorizes the control command which should perform the aforementioned processor, When the control command memorized by this command storage section and the aforementioned print data transmitted to the aforementioned buffer memory by the direct-memory-access method are in agreement, An address storage means to memorize the address of the aforementioned buffer memory these print data are remembered to be, With reference to the print data memorized by the aforementioned buffer memory, the print data of the address memorized by the aforementioned address storage means are read, and it is characterized by having a control execution means to perform processing according to the aforementioned control command.

[0010]

[Function] When the control command memorized by the command storage section and the print data of the printer buffer of this invention constituted as mentioned above transmitted to buffer memory by the direct-memory-access method correspond, an address storage means memorizes the address of the buffer memory the print data is remembered to be. And a control execution means reads the print data of the address memorized by the address storage means with reference to the print data memorized by buffer memory. Since this print data is control command, processing according to this control command is performed.

[0011] In addition, even if address information is absolute-address information on buffer memory, any of the relative address information embellished by the BEJI information on DMA processing etc. are sufficient as it here.

[0012] Moreover, the general data retrieval technique, such as detecting not only the full coincidence with the control command and print data which were memorized by the command storage section but the coincidence with information, such as information common to a control command system, for example, a command head etc., and print data, is suitably applied to detection of control command.

[0013]

[Example] In order to clarify further the composition of this invention explained above, and an operation, the suitable example of the printer buffer of this invention is explained below. Drawing 1 is the electrical circuit block diagram of the printer buffer 20 which is an example. A printer buffer 20 is constituted by the logical circuit centering on a microprocessor (henceforth CPU)

22, it consists of ROM24 which memorizes data, such as a program mentioned later, in un-volatilizing, RAM26 which becomes together with this ROM24 and serves as a primary storage of CPU22, various I/O circuits, etc., and these circuits are mutually connected by internal bus B so that it may illustrate. In addition, this internal bus B is a signal line to which device selection, input data, output data, and the various signals of appliance control are transmitted as a memory-related signal as the address, read data, write-in data, the various signals of memory device control, and an I/O circuit-related signal. In addition, it does not interfere that such a bus takes the composition to which even ** transmits these signals for the increase in a signal line to a ** sake by time sharing, either.

[0014] The data input circuit 30 is an input/output interface which delivers and receives print data between computers 10, changes the print data which follow various kinds of printer telecommunications standards, and are transmitted from a computer into the data format which can treat CPU22, and outputs them to the internal bus B to timing suitably. The data output circuit 32 changes the data processed by CPU22 into the data format which suits various kinds of printer telecommunications standards contrary to operation of the above-mentioned data input circuit 30, and outputs them to a printer 15. Moreover, buffer memory 34 consists of mass storage elements, accumulates temporarily the print data inputted from the data input circuit 30, and outputs this to the data output circuit 32 suitably.

[0015] The printer buffer 20 of an example was faced memorizing the print data inputted from the data input circuit 30 to buffer memory 34, performed well-known DMA processing, and has improved data transfer speed sharply. In order to control this DMA processing, DMA controller 40 is connected to the above-mentioned internal bus B.

[0016] DMA controller 40 of this example controls the internal bus B by the so-called steel mode. That is, if DMA processing is needed, CPU22 will set up the size of a DMA transfer byte count (an example 4 K bytes) and a storage page, the transfer starting address of buffer memory 34, etc. as information required for a demand and data transfer of a DMA transfer. DMA controller 40 transmits after that the print data inputted from the data input circuit 30 according to the transfer byte count and transfer starting address which are beforehand set up from the data input circuit 30 using the period when CPU22 is not using the internal bus B according to a print entry of data to buffer memory 34.

[0017] Furthermore, the printer buffer 20 of this example is equipped with the comparator circuit 42 and the address latch circuit 44. In a comparator circuit 42, when the criteria data set from CPU22 are compared with the data transmitted to buffer memory 34 by DMA processing and both data are in agreement, it is the circuit which outputs a latch trigger signal to the address latch circuit 44. The address latch circuit 44 which inputs this latch trigger signal is a circuit which latches the address information of the internal bus B when a latch trigger signal is inputted, and memorizes the address information with 16 FIFO (henceforth FIFO) methods. Moreover, the overflow terminal OF of the address latch circuit 44 is connected to the interruption terminal of the above CPU 22, and the DMA halt terminal of DMA controller 40, and when 16 or more address information exceeding storage capacity is inputted, while outputting a signal from this overflow terminal OF and stopping DMA processing, CPU22 is made to execute an overflow interruption program. In addition, in anticipated use, even if there is much generating frequency of the control command in printing data, possibility is a low very much and accumulating 16 or more commands with unsettled CPU22 hardly produces that FIFO overflows. In addition, what is necessary is just to prepare FIFO further, when much control command is predicted.

[0018] The various programs like the following are beforehand memorized by ROM24, and the printer buffer 20 of this example constituted as mentioned above operates as follows. In case drawing 2 requires the DMA processing whose CPU22 are the flow chart of a DMA demand and an executive program, and used the internal bus B for DMA controller 40, it is performed during execution of a DMA transfer. If processing of this program is started, the DMA processing after clearing DMA field use of the address latch circuit 44 or buffer memory 34 will be equipped with CPU22 (Step 100).

[0019] Then, CPU22 sets 2 bytes of data "@B" as criteria data to a comparator circuit 42 (Step

110). Here, 2 bytes of data "@B" are the common data added to the head of the various commands to the printer buffer 20 of this example, and data of the so-called command head. That is, to the printer buffer 20 of this example, print number of copies is specified or the control command of specifying the printer which wishes print work is described after the 2-byte data "@B" of a command head.

[0020] In this way, when all preparations for DMA processing are completed, CPU22 points to a transfer byte count, the transfer starting address of buffer memory 34, etc. as initial-setting information for DMA to DMA controller 40 (Step 120), and makes the DMA transfer in steel mode start. If a DMA transfer starts, although CPU22 performs 130 or less-step processing, it will be explained in detail after explanation of operation by the DMA transfer about this processing.

[0021] If DMA controller 40 starts DMA processing based on a DMA demand, the DMA transfer by the usual steel mode will be performed, and the print data inputted from the data input circuit 30 will be asynchronously transmitted to the direct buffer memory 34 with CPU22. And when the data which are in agreement with the command head "@B" set by the comparator circuit 42 into the print data by which a DMA transfer is carried out in the midst of this DMA processing appear in the internal bus B, a latch trigger signal is outputted to the address latch circuit 44 from a comparator circuit 42, and the address information of the buffer memory 34 the command head "@B" was remembered to be is memorized one by one by the address latch circuit 44 with a FIFO method. That is, the address information of the buffer memory 34 the command head "@B" was remembered to be is completely accumulated in hard at the address latch circuit 44, without making CPU22 completely intervene. In addition, what is necessary is to memorize this to the predetermined address and just to refer to later, in case the address of a high order is set up, when the address information in a DMA transfer serves as a local address in a transfer page. It is good also as composition which combines the high order address with FIFO and memorizes it from the first.

[0022] While the above processing is performed by the DMA transfer, CPU22 judges whether data are set to FIFO (Step 130). If FIFO is not empty, the data of FIFO, i.e., the address data with which the software command is saved, will be read (Step 140), and the data of the address will be read (Step 150). Next, this data is analyzed and it judges whether it is the command which should be processed by the input side of a buffer (Step 160), and in being data which should be processed by the input side, it executes this control command (Step 170). The processing which should be performed by the input side, and the processing which should be performed by the output side exist in a software command. The processing which should be performed by the input side is processing which clears immediately the contents of a buffer including the data for example, under present printing. Processing of the clearance of a buffer clearing the break pointer of data or subsequent ones also exists. In this case, what is necessary is just to save the previous software command to the predetermined field of a buffer. **** [the processing which should be performed by the output side / the printer which is for example, a printing place] -- ** -- *****

[0023] Especially when data are not set to FIFO (Step 130) and the set command is not a command which should be executed by the input side after executing control command or (Step 160), control command is not processed, it judges whether the DMA transfer was completed (Step 180), and 130 or less-step processing is repeated during continuation of a DMA transfer.

[0024] Next, the processing which prints out the printing data accumulated to buffer memory 34 is explained using the flow chart of drawing 3. If processing of this program is started, CPU22 will check a series of print data storage fields stored in buffer memory 34 by DMA processing (Step 200). The start address which reads data is set to the address counter of CPU22 by this processing. Then, processing which reads data from the field directed to the address counter is performed (Step 205).

[0025] Then, CPU22 judges whether the read data are a software command (Step 210). When it was not a software command, i.e., it is judged that it is printing data, the usual print processing which outputs the data to the data output circuit 32 is performed (Step 220).

[0026] On the other hand, when the data read at Step 210 are judged to be a software

command, it judges whether the command is a command which should be processed by the output side (Step 225). In being the command which should be processed by the output side, for example, the change command of the printer which is a printing place, it performs control according to the command (Step 230).

[0027] In this way, if processing of the stored data of the buffer memory 34 directed in the address counter is completed, the content of an address counter is incremented (Step 240), and it will return to the aforementioned step 205 until the same processing is completed about all data read-out ranges.

[0028] Thus, the data by which the DMA transfer was carried out to buffer memory 34 are analyzed one by one, and in the case of the software command which requires execution at the time of an output, corresponding processing is performed, and the other data are judged to be printing data and are outputted to a printer from the data output circuit 32. Therefore, although the information outputted to a printer by DMA processing and the information on control command are intermingled and it is stored in buffer memory 34, in read-out of the mixture data, control command can be separated and extracted simply and certainly, and it becomes possible to perform control according to the control command inputted through the data input circuit 30 from the computer 10 by CPU22.

[0029] Drawing 4 is the flow chart of the interruption program executed when overflow occurs in the address latch circuit 44 which plays an important role in this example like the above and interruption occurs to CPU22. In addition, possibility that possibility will be a low very much and, as for CPU22 accumulating 16 or more commands while it has been unsettled, overflow will occur like this example in the address latch circuit 44 which can latch 16 address data by the generating frequency of the control command contained in the data usually inputted from a computer is a low very much.

[0030] Like the above-mentioned, the address latch circuit 44 is demanding a halt of DMA processing from DMA controller 40 simultaneously with this overflow interruption, and control of the internal bus B is left to CPU22. Then, CPU22 checks a print entry-of-data situation to the data input circuit 30, it judges whether print data are yet transmitted from the computer, and transfer of print data checks whether the handshake performs or not (Step 300). That is, when transfer of the print data between a computer 10 and a printer buffer 20 is performed without the handshake, the situation that print data are independently transmitted to a target on the other hand from a computer is continuing a halt of DMA processing. On the other hand, when transfer of the print data between a computer and a printer buffer 20 is performed by the handshake, the print data input to the data input circuit 30 is interrupted by computer-processing.

[0031] then, when it judges that the existence of a handshake was judged by the following step 310, the handshake was taken, and transfer of print data was performed Future print entries of data are performed by the program control between which it was placed by CPU22, without performing DMA processing. Print data are memorized to buffer memory 34, detecting the information and distributing to the predetermined field of RAM26, when the command head "@B" is contained in the print data inputted (Step 320).

[0032] On the other hand, since the control command taken and spilt by the input side is analyzed by the program by the side of printing data output (drawing 3) even when print data are inputted without the handshake at Step 310, the delay of the response to a command does not produce fault in execution of the printing data of a certain thing, or a command.

[0033] If it is in the handshake state which was mentioned above and which does not have the possibility of problem generating in transfer of print data though it interrupts and overflow of data generates a printer buffer 20 in the metaphor address latch circuit 44 by the program, it will become possible to receive future print data. Moreover, a printer buffer 20 can output that to a computer, when there is possibility of overrun of print data, and it can stop the print entry of data automatically.

[0034] According to the printer buffer of this example, as explained above, or since 2 bytes of data comparison is performed by high-speed hard processing and control command is detected, the control data included in print data can be separated and processed certainly and correctly

by DMA processing and synchronization. Therefore, it can be satisfied [with executing separated control command certainly] of the demand of the advanced features which operate a printer buffer 20 by remote control not to mention inputting print data into buffer memory 34, and DMA processing of the maximum high speed being realized, without interrupting the once started DMA processing. That is, a printer buffer 20 can input print data, without performing program control which needs the considerable processing time like interrupt processing of CPU22, and when print data are inputted by high-speed serial communication without a handshake, it can respond with sufficient speed of response.

[0035] Moreover, even if it is the case where the address information temporarily memorized by the address latch circuit 44 overflows, a print entry of data can be continued and useless data communication which is not can be realized.

[0036] In addition, although the command system of control command was unified and 2 bytes of command head "@B" is detected in a comparator circuit 42 in this example in order to simplify and accelerate processing, the control command of the another kind which belong and is not in such a command system can also be extracted by preparing two or more such comparator circuits.

[0037] Moreover, in order to simplify processing when overflow of address information occurs in the address latch circuit 44 in such a case, DMA processing may always stop, and you may transmit the message which tells overrun generating of print data to a computer. In this example, although a part of analysis of control command was performed at the time of a DMA transfer, carrying out collectively, after the DMA transfer of printing data is completed, and attaining simplification of analysis does not interfere, either.

[0038] Although the example of this invention was explained above, as for this invention, it is needless to say that it is embodied by the mode which is not limited to such an example at all and does not deviate from the summary of this inventions, such as what the header of the composition and control command using DMA transfers other than steel mode equips with 3 bytes or more of composition, and which becomes various.

[0039]

[Effect of the Invention] When the control command by which the printer buffer of this invention was remembered to have explained above by the command storage section, and the print data transmitted to buffer memory by DMA system are in agreement, an address storage means memorizes the address of the buffer memory the print data is remembered to be. And with reference to the print data memorized by buffer memory, a control execution means reads the print data of the address memorized by the address storage means. Since it can judge that this print data is control command, processing according to the control command is performed. Therefore, not to mention storing print data in the maximum high speed by DMA processing at buffer memory, the printer buffer of this invention performs complicated processing according to control command certainly, and functions as high efficiency and a printer buffer of high-speed processing.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the electrical circuit block diagram of the printer buffer which is one example of this invention.

[Drawing 2] It is the flow chart of the DMA demand program performed in CPU of the printer buffer.

[Drawing 3] It is the flow chart of the print program performed in CPU of the printer buffer.

[Drawing 4] It is the flow chart of the interruption program executed in CPU of the printer buffer.

[Description of Notations]

10 -- Computer

15 -- Printer

20 -- Printer buffer

22 -- CPU

24 -- ROM

26 -- RAM

30 -- Data input circuit

32 -- Data output circuit

34 -- Buffer memory

40 -- DMA controller

42 -- Comparator circuit

44 -- Address latch circuit

B -- Internal bus

OF -- Overflow terminal

[Translation done.]